

# Gestió d'una base de dades amb informació geo-posicionada

Andreu Ros Gutiérrez

**Resum**—Cada vegada és més gran el nombre d'aplicacions que utilitzen informació geo-posicionada per donar nous serveis als seus usuaris. Aquesta informació cal emmagatzemar-la en una base de dades i ser tractada de forma adequada. En aquest context s'emmarca l'aplicatiu web desenvolupat en aquest projecte, que tracta de gestionar la base de dades de l'aplicació Incidències 2.0, per a dispositius Android, que tracta d'informar als usuaris de les incidències que tenen lloc en el transport públic de l'àrea metropolitana de Barcelona i els permet localitzar T-11. L'aplicació permet als administradors d'Incidències 2.0 gestionar aquesta base de dades i visualitzar de forma gràfica la informació continguda, a més d'afegir, modificar, eliminar incidències i T-11 i generar informes de seguiment. El projecte s'ha desenvolupat seguint una metodologia de desenvolupament de software basada en un model de prototipatge, separant l'aplicatiu en tres grans mòduls funcionals i independents: gestió d'incidències, gestió de les T-11 i generació d'informes, seguint el patró de disseny Model Vista Controlador i utilitzant diferents llenguatges i tecnologies que permeten el tractament d'informació geo-posicionada.

**Paraules clau**—Incidències, transport públic, àrea metropolitana de Barcelona, geo-localització, informació geo-posicionada.

**Abstract**—The number of applications that use geo-positioned information to give new services to their users is increasing every day. This information has to be stored in a data base and has to be processed adequately. The web application developed in this project is framed in such scenario and provides a tool for managing the remote database of Incidències 2.0, an Android application that allows users to report and stay informed about incidents in the public transportation of the metropolitan area of Barcelona, as well as locating T-11 cards. The application allows the managers of Incidències 2.0 manage this database and visualize, in a graphical way, the stored information, as well as insert, update and delete incidents and T-11 and generate monitoring reports. The project has been developed following a software developing methodology based on a prototyping model, dividing the application in three functional and independent modules: incident management, T-11 management and report generation, following the Model View Controller design pattern and using different languages and technologies that allows the processing of geo-positioned information.

**Index Terms**—Incidents, public transport, Barcelona metropolitan area, geo-location, geo-positioned information.



## 1 INTRODUCCIÓ

AQUEST projecte s'emmarca dins del projecte Incidències 2.0 [1], un sistema que pretén donar solució a la manca d'informació o a la informació no actualitzada sobre l'estat de la xarxa de transport públic.

El projecte Incidències 2.0 consta d'una aplicació mòbil per dispositius mòbils amb sistema operatiu Android que dona la possibilitat als usuaris de reportar i consultar diferents tipus de incidències que es poden donar en el transport públic de l'àrea metropolitana de Barcelona. (Metro, Ferrocarrils de la Generalitat o Rodalies Renfe). A més a més, l'aplicació inclou una funcionalitat que permet als usuaris compartir l'últim viatge dels títols TMB T-10 sempre que aquest es transfereixi a una altra persona dins del temps màxim permès de transbord. Aquesta iniciativa

es coneix com T-11 i l'aplicació d'Incidències 2.0 permet informar sobre el lloc on es deixen les T-11 o cercar-ne d'existents.

Es tracta, doncs, d'una aplicació col·laborativa on són els propis usuaris qui comuniquen als altres usuaris de l'aplicació les diferents incidències que estan ocorrent. Tot i això, per facilitar el procés de validació automàtica, tota la informació relacionada amb l'aplicació es guarda en una base de dades remota.

L'objectiu d'aquest projecte és desenvolupar una aplicació web per facilitar el manteniment i explotació de la base de dades Incidències 2.0. L'aplicació, doncs, ha de permetre als administradors de la base de dades d'Incidències 2.0 consultar i gestionar la informació enviada pels usuaris de l'aplicació de forma eficient, així com modificar, afegir i eliminar incidències de forma ràpida i senzilla i generar informes periòdics que continguin informació i estadístiques sobre la base de dades.

A més, s'ha de restringir l'accés a la gestió i consulta de la informació de la base de dades de Incidències 2.0 als

---

- E-mail de contacte: andreurosg@gmail.com
- Menció realitzada: Tecnologies de la Informació.
- Treball tutoritzat per: Cristian Tanas (dEIC)
- Curs 2013/14

administradors, assegurant la integritat, la privacitat i la disponibilitat de les dades.

Aquest article està estructurat de la següent manera. En la Secció 2 es presenta la problemàtica que sorgeix del desenvolupament del projecte Incidències 2.0 i les tecnologies actuals existents per dur a terme un projecte d'aquestes característiques. La metodologia seguida pel desenvolupament de l'aplicatiu web s'exposa en la Secció 3. La Secció 4 recull els resultats i les proves realitzades al llarg del desenvolupament del projecte. Finalment, la Secció 5 presenta les conclusions i futures línies de treball.

## 2 ESTAT DE L'ART

Donat que en aquest projecte es desenvoluparà un aplicatiu web i s'enmarca dintre del projecte Incidències 2.0, aquests són els principals punts a tractar en aquesta secció.

### 2.1 Incidències 2.0

L'aplicació per a smartphones Incidències 2.0 està operativa i funciona connectant-se a una base de dades on es guarda la informació de les incidències que reporten els usuaris. Tanmateix, actualment els administradors gestionen aquesta base de dades executant sentències SQL directament i consultant informes generats automàticament de forma periòdica per un script.

Això fa que el manteniment i explotació d'aquesta base de dades sigui poc eficient i a mesura que la base de dades augmenti de mida es farà del tot inviable. Així, apareix la necessitat de desenvolupar una solució que permeti gestionar la informació d'aquesta base de dades de forma còmoda pels administradors, que sigui escalable i que faci transparent l'accés a la base de dades.

### 2.2 Tecnologies de desenvolupament web

En l'actualitat existeixen moltes aplicacions que han de consultar i gestionar informació geo-posicionada. Des d'utilitats per a dispositius mòbils com els GPS o mapes fins a aplicacions per compartir la nostra ubicació com FourSquare. I no només ens ho trobem en software on la geo-localització es indispensable, altres tipus de aplicacions com Facebook o Whatsapp utilitzen geo-localització per donar nous serveis als seus usuaris.

Una de les claus ha estat el creixent ús de Google Maps, que ha obligat a casi totes les aplicacions a posar-se al dia i utilitzar la geo-localització per millorar l'experiència de l'usuari amb l'aplicació. Google Maps a més ofereix una API oberta que facilita als desenvolupadors el tractament d'informació geo-posicionada i que funciona conjuntament amb JavaScript.

Per poder desenvolupar una interfície senzilla i intuïtiva i presentar la informació de forma gràfica a l'usuari, actualment les tecnologies d'ús més comú en el marc de les aplicacions web són la llibreria jQuery i el conjunt d'eines de Twitter Bootstrap, les dues tecnologies de codi obert en JavaScript més utilitzades del moment, junta-

ment amb HTML5 i CSS3, les últimes versions dels dos llenguatges.

Avui en dia, quan parlem d'un aplicatiu web no parlem d'una pàgina web tradicional amb continguts estàtics, sinó que parlem de webs dinàmiques on un dels llenguatges més habituals en la banda del servidor és PHP, per la seva condició de software lliure, multiplataforma (funciona en pràcticament tots els servidors web i sistemes operatius) i multiparadigma (es pot utilitzar com a llenguatge orientat a objectes o com a llenguatge procedural). Per a la comunicació amb la base de dades és habitual utilitzar llenguatges com MySQL, ja que és perfectament compatible amb PHP.

Actualment, també existeixen moltes llibreries de codi obert desenvolupades per a PHP, que permeten donar solució a les dificultats per implementar algunes funcionalitats. Una d'aquestes llibreries és TCPDF, que proporciona mètodes que permeten generar i escriure arxius de text en format PDF.

## 3 METODOLOGIA

A continuació s'exposarà la metodologia que s'ha seguit per dur a terme el projecte, fent èmfasi en les fases d'anàlisi, disseny i implementació.

### 3.1 Desenvolupament de l'aplicatiu web

S'ha seguit una metodologia de desenvolupament de software basada en un model de prototipatge [2] [3], en el qual s'han anat incorporant, de forma escalonada, mòduls amb diferents funcionalitats al software.

Aquest model consisteix en realitzar un disseny relativament ràpid del producte i construir-lo àgilment per tal d'entregar-lo al client (en el nostre cas, el tutor del treball) per a que l'avalui. Aquest procés permet retroalimentació entre el client i el desenvolupador, ja que el client rep informació de com s'està construint el producte que es desitja i el desenvolupador pot captar nous requisits que havien passat per alt. Aquest model, doncs, és especialment eficaç quan es tenen clars els objectius generals del projecte però no s'identifiquen requisits detallats. La presentació de prototipus al client permet que aquest comuniqui explícitament aquests requisits no identificats prèviament. La figura 1 mostra l'esquema bàsic d'aquest model:

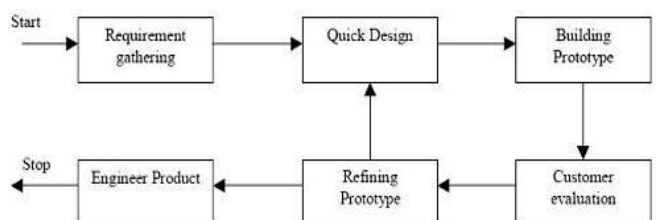


Figura 1: Esquema del model de prototipatge (Font: [www.freetutes.com](http://www.freetutes.com) – Visual Basic Tutorials)

Es va escollir el model de prototipatge per desenvolupar aquest projecte perquè a partir de les reunions amb el tutor es deixaven clar els objectius generals de cada mòdul, i a partir de les entregues de prototipus es podien el·licitar requisits més detallats, tant de la interfície d'usuari com del tractament de certes dades en els diferents mòduls del sistema.

Per cada mòdul desenvolupat, es van aplicar les fases de l'enginyeria del software (anàlisi de requisits, disseny, implementació i proves). A partir dels requisits s'han definit els casos d'ús pertinents i s'han identificat escenaris per cada cas d'ús. Abans de la implementació s'han definit quines classes i mètodes seran necessaris i s'han fet esbossos a mà de la interfície d'usuari.

### 3.2 Anàlisi de requisits

Com s'ha explicat en la secció anterior, seguint el model de prototipatge, a partir de reunions amb el tutor es definien els objectius generals, i en base a aquests objectius es dissenyaven i es construïen prototipus. A partir de l'avaluació de prototipus s'identificaven els requisits detallats del software amb els quals es refinava el prototipus fins que aquest satisfia tots els requisits. Amb els requisits definits, s'identificaven i es detallaven els diferents casos d'ús (veure Annex A1 per consultar el diagrama de casos d'ús). Per cada cas d'ús s'assigna un ID únic que serveix per referenciar de quins requisits es deriven i, per tant, mantenir la traçabilitat amb els requisits i amb el seu nom. S'identifiquen els actors que hi intervenen, la prioritat (que pot ser alta, mitjana o baixa), les pre-condicions i post-condicions per a que s'executi el cas d'ús, altres casos d'ús utilitzats, el flux d'events detallat i un llistat d'escenaris alternatius (un escenari alternatiu típic, seria que es produís un error de connexió amb la base de dades durant l'execució del cas d'ús).

Els requisits es van classificar en funcionals i no funcionals i els funcionals es van separar per mòduls.

#### 3.2.1 Requisits funcionals

Els administradors han de poder visualitzar les incidències de dues formes diferents: en un mapa i en forma de llistat. Es distingeixen tres tipus d'incidències:

- Incidències de transport públic.
- Incidències generals.
- Incidències obtingudes per RSS.

Les incidències de transport públic i generals són enviades pels usuaris de l'aplicació Incidències 2.0 i contenen informació de posicionament. De cada tipus d'incidència es guarda diferent informació, alguna comuna entre els diferents tipus i alguna altra específica.

En el llistat d'incidències han d'apareixer la descripció, l'estat (incidència confirmada, sense confirmar o caducada) i la data d'enviament. En les incidències de transport i les obtingudes per RSS, a més a més, han d'apareixer el servei (Rodalies, FGC o Metro) i la línia on s'han produït.

En el mapa, s'ha de diferenciar cada incidència utilitzant un marcador diferent per cada tipus. Si es clica en el marcador, ha d'apareixer un quadre de diàleg amb la

informació de la incidència. Les incidències obtingudes a través de fonts RSS només estaran disponibles per llistat.

Quan s'accedeixi a una incidència, s'haurà de mostrar tota la informació relativa a la incidència, incloent la que ja es mostrava en el llistat (latitud, longitud, severitat, data de modificació, usuari que l'ha enviat, TTL – *Time To Live*, estació en que s'ha produït, direcció en la que anava el tren i causa, entre d'altres). Cada informació s'ha de mostrar en un camp que ha de ser editable.

Els administradors han de poder modificar i eliminar qualsevol incidència, així com afegir manualment incidències de transport i generals especificant tota la informació relativa a la incidència.

S'han de poder filtrar incidències per tipus, estat, causa, severitat, servei i línia, tant en el llistat com en el mapa, podent aplicar tants filtres a la vegada com es vulgui.

De la mateixa manera que amb les incidències, l'usuari ha de poder visualitzar les T-11 en un llistat i en un mapa a part. En el llistat apareixerà la descripció, l'estat, la línia, la data d'enviament i la data de recollida de la T-11, i s'han de poder filtrar per l'estat.

També, s'haurà de poder accedir a cada T-11, veure tota la informació relativa a ella i poder-la modificar o eliminar. A més, s'han de poder afegir T-11 manualment.

Finalment, l'usuari haurà de poder generar informes de seguiment a partir d'una data d'inici. Aquests informes han de ser generats en format PDF i han de contenir informació estadística sobre el nombre d'incidències rebudes en els diferents serveis de transport públic, el nombre d'incidències generals, el nombre de T-11 deixades així com informació relativa a les incidències obtingudes a través dels canals RSS.

#### 3.2.2 Requisits no funcionals

El sistema ha de mostrar la informació organitzada de forma que la navegació resulti intuïtiva (sense necessitat de consultar cap manual ni cap ajuda) i la interfície ha de ser senzilla i poc cridanera.

L'aplicació ha de poder ser accessible des de diferents sistemes operatius i plataformes. Cal restringir l'accés a tota l'aplicació als responsables d'Incidències 2.0, i per tant s'ha d'assegurar que cap altra persona hi pot accedir.

A més, el sistema ha de ser ràpid, robust i escalable, permetent la manipulació de molta informació a la vegada.

En quant a la seguretat, cal assegurar que només puguin accedir a l'aplicació i a qualsevol dels recursos de l'aplicatiu aquells usuaris que demostrin tenir credencials d'accés. Així mateix, cal protegir l'aplicació contra possibles errors i atacs que puguin posar en perill la integritat, la disponibilitat i la privacitat de les dades.

### 3.3 Disseny

#### 3.3.1 Mòduls funcionals de l'aplicació

Donat l'anàlisi de requisits anteriors, es pot separar l'aplicació en quatre grans mòduls, que poden funcionar de forma independent i que treballen per separat, com es mostra en la Figura 2:

- *Mòdul d'accés a l'aplicació*: Controla que només accedeixin a l'aplicació els usuaris pertinents.
- *Mòdul de gestió d'incidències*: Gestiona les incidències de transport, incidències generals i incidències obtingudes mitjançant fonts RSS. Permet veure un llistat de totes les incidències, situar-les en un mapa i filtrar-les segons el tipus d'incidència, el servei, la línia, l'estat i la severitat, així com afegir, modificar i eliminar incidències.
  - Submòdul de visualització d'incidències
  - Submòdul d'inserció d'incidències
  - Submòdul de modificació i eliminació d'incidències
- *Mòdul de gestió de les T-11*: Gestiona les T-11 reportades pels usuaris. Permet veure un llistat de les T-11 reportades i situar-les en un mapa, així com afegir-ne de noves, modificar-les i eliminar-les.
- *Mòdul de generació d'informes*: Permet generar informes que donen informació estadística sobre l'ús de l'aplicació Incidències 2.0 i les incidències succeïdes en un cert període de temps.

Tota l'aplicació s'ha separat tenint en compte aquests mòduls, de manera que en realitzar canvis sobre un mòdul o fins i tot eliminar un dels mòduls no canviï el funcionament de la resta de mòduls.

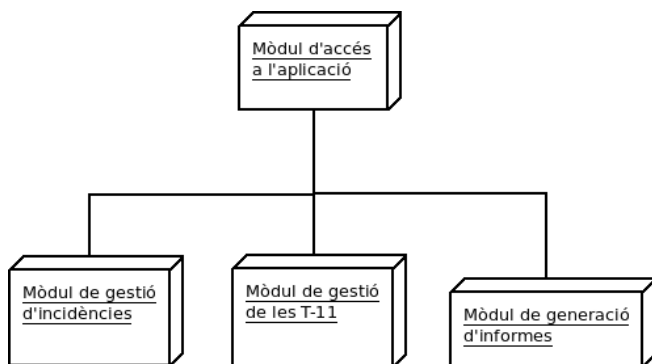


Figura 2: Diagrama de mòduls de l'aplicació

Els mòduls desenvolupats poden compartir classes del model, ja que el model es un conjunt de classes i mètodes que poden ser utilitzats des de qualsevol lloc de l'aplicació. Fins i tot, podria aprofitar-se el model i ser utilitzat en un aplicatiu diferent que necessiti explotar la mateixa base de dades.

### 3.3.2 Arquitectura del software

L'aplicació s'ha desenvolupat seguint el patró de disseny Model Vista Controlador [4], que separa les dades i la lògica de negoci de l'aplicació de la interfície gràfica amb la qual interacciona l'usuari.

S'ha decidit utilitzar aquest patró d'arquitectura software, ja que el sistema treballa constantment consultant informació de la base de dades d'Incidències 2.0.

Aquesta informació es visualitza en diferents formats (llistat d'incidències, mapa, pàgina d'informació d'una incidència o informes). Per tant, es necessiten les mateixes dades en diferents vistes. El patró Model Vista Controlador ens permet utilitzar la mateixa lògica de negoci en diferents interfícies gràfiques, a més de realitzar canvis sobre la lògica de negoci o la interfície sense haver de modificar ambdues coses, tal i com es pot veure en l'esquema de la Figura 3.

En aquesta aplicació, el model engloba tota la lògica que s'encarrega de realitzar consultes a la base de dades, tractar les dades consultades per extreure'n informació en el format adequat i realitzar operacions sobre la base de dades per afegir, modificar i eliminar informació. Per tant tot el model està programat en PHP [5] i les consultes i operacions sobre la base de dades es realitzen amb sentències MySQL. Les vistes s'encarreguen d'organitzar i estructurar la informació en la pàgina i d'aplicar estils. Per tant, contenen tota la part relacionada amb la interfície gràfica, que s'encarrega de presentar la informació a l'usuari (això es tot el codi HTML, CSS i JavaScript). Els controladors són el punt d'entrada a l'aplicació i als seus recursos. S'encarreguen de fer d'intermediaris entre el model i la vista. Quan l'usuari demana un recurs, el controlador li demana al model que faci les transaccions corresponents, per després enviar a la vista la informació obtinguda del model, per a que la presenti en el format adequat. Els controladors han estat programats únicament en PHP.

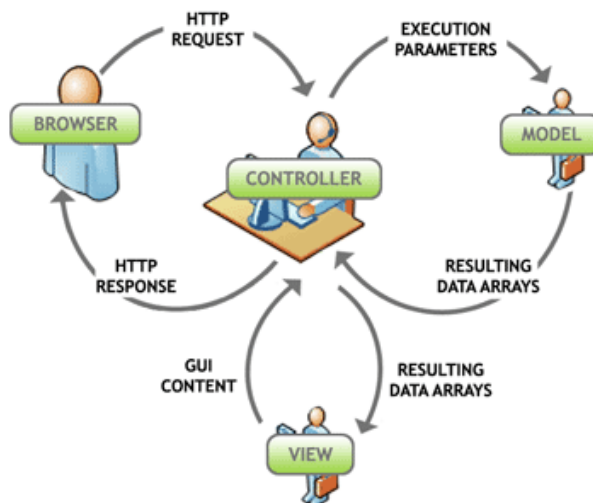


Figura 3: Esquema del patró Model Vista Controlador (Font: [www.developer.chrome.com](http://www.developer.chrome.com))

### 3.3.3 Disseny de la interfície gràfica

Abans de començar a implementar, es van fer esbossos a mà de com hauria de ser la interfície de l'aplicació. Aquests esbossos van ser ensenyats al tutor del projecte, i un cop aplicades certes millores en el disseny suggerides pel tutor, es va tornar a presentar l'esbòs per a que finalment fos validat.

Un cop validat l'esbòs fet a mà, es va començar a im-

plementar la interfície (sense cap funcionalitat, és a dir, utilitzant només HTML + CSS + JavaScript). Un cop implementada, es va començar a incloure les funcionalitats desenvolupades en la interfície.

La idea principal va ser organitzar les diferents opcions de l'aplicació en *tabs*. Un *tab* és un element utilitzat en interfícies gràfiques que permet incloure, en una sola finestra, diferents documents o panells. Actualment, Per canviar de panell o de document només cal clicar en el *tab* desitjat. Així, en la pantalla principal de l'aplicació pot aparèixer el panell d'incidències, el panell de T-11 o el de generació d'informes segons el *tab* que cliquem. Dins del panell d'incidències i T-11, també podem escollir mostrar el llistat, el mapa o el formulari per afegir una incidència, tal i com es mostra en la Figura 4.



Figura 4: Captura del panell d'incidències amb els tabs disponibles

Una altra de les bases sobre les que està construït el disseny de la interfície gràfica de l'aplicació, és la idea d'organitzar la informació en panells. Així, per exemple, dins de l'opció d'incidències hi ha dos panells: un panell amb els filtres que es poden aplicar i un panell on es mostren les incidències.

Com que es va creure que aquest tipus de disseny era el més pràctic i intuïtiu per un aplicatiu web, es va decidir utilitzar la llibreria Twitter Bootstrap, que incorpora tota la funcionalitat necessària per implementar aquests elements en la interfície gràfica.

### 3.4 Implementació

Per a la visualització d'incidències (tant en el mapa com en el llistat), ha estat necessari desenvolupar un sistema de filtratge per facilitar-ne la visualització, tal i com demanaven els requisits del sistema, ja que a la base de dades hi ha un gran volum d'incidències que a més pot fer-se molt més gran amb el temps. Els filtres permeten reduir el nombre d'incidències en el llistat i en el mapa i trobar més fàcilment les incidències desitjades. Els filtres que s'han implementat són els següents:

- Tipus d'incidència
- Estat
- Severitat
- Servei
- Línia
- Causa

Poden aplicar-se diversos filtres a la vegada, tenint en compte que per alguns tipus d'incidència no es podran utilitzar tots els filtres. Quan l'usuari seleccioni un filtre el sistema actualitzarà la vista de filtres:

- Filtre per incidències de tipus general: Es deshabilitaran els filtres de servei, línia i causa.
- Filtre per incidències de tipus RSS: Es deshabilitaran els filtres de causa i severitat.

Cal afegir que existeixen dependències entre filtres i que aquestes també es tenen en compte. Per exemple, si es canvia la línia, canvia automàticament el servei. Al contrari, si es canvia el servei, es deixa de filtrar per línia. Totes aquestes dependències s'apliquen utilitzant AJAX (*Asynchronous JavaScript And XML*) [6], ja que permet recarregar només la vista corresponent als filtres sense necessitat de recarregar tota la pàgina, demanant les dades al servidor mitjançant l'objecte *XMLHttpRequest* [7] sense interferir ni en la visualització ni en el comportament de la pàgina. La Figura 5 mostra de forma esquemàtica el funcionament d'AJAX.

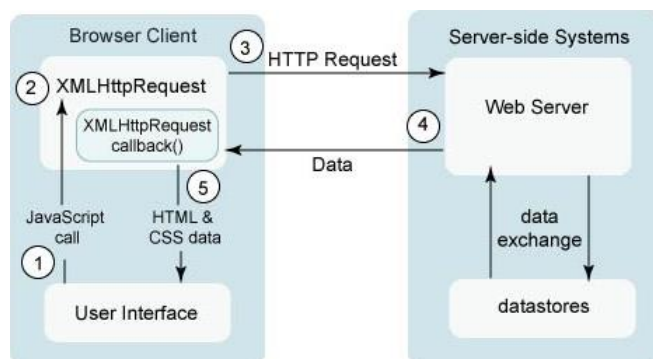


Figura 5: Esquema de la transferència de dades mitjançant AJAX (Font: [www.courses.cs.washington.edu](http://www.courses.cs.washington.edu))

En general, la tecnologia AJAX ha estat clau pel desenvolupament de tota l'aplicació, ja que permet fer la navegació molt més dinàmica en aplicacions web com aquesta, on la informació i els continguts a mostrar canvien contínuament. A més, ha estat especialment important en el sistema de filtratge, ja que permet a l'usuari aplicar filtres de forma molt més dinàmica (cada cop que es selecciona un filtre s'actualitza automàticament el llistat o el mapa).

Això es fa des de la vista, enviant una petició AJAX amb els filtres seleccionats per l'usuari al controlador del llistat, per a que aquest actualitzi la vista. El controlador cridarà al gestor d'incidències, que carregarà de la base de dades totes les incidències que concordin amb els filtres i retornarà al controlador un array d'objectes de la classe Incident. El controlador recarregarà la vista del llistat.

En el cas del mapa, el sistema ha de carregar la informació de les incidències de la base de dades, generar una seqüència XML amb la informació que han de contenir els marcadors (*markers*), enviar-lo utilitzant també AJAX a la vista corresponent al mapa i situar els markers en el mapa a partir de la latitud i la longitud.

En l'Annex A2 podem veure el diagrama de classes de l'aplicatiu. El disseny de classes s'ha desenvolupat tenint en compte el disseny de la base de dades d'Incidències 2.0. Les classes TransportIncident, GeneralIncident i RSSIncident contenen tota la informació relativa a cada



tipus d'incidència i proveeixen mètodes per afegir, modificar i eliminar incidències i per traduir certa informació que s'emmagatzema en la base de dades amb un identificador (com el servei de transport, la línia, la causa de la incidència, etc.). Donat que `TransportIncident` i `GeneralIncident` comparteixen molts atributs i mètodes, ambdues classes hereden de la classe `Incident`, on es defineixen els atributs i mètodes comuns a les dues classes.

La classe `TransportIncident`, així com la classe `GeneralIncident` guarden atributs que en la base de dades s'emmagatzemen en més d'una taula, per tant, a l'hora de desenvolupar aquestes classes es va haver de tenir molt present el disseny de la base de dades `Incidències 2.0`, per tal de realitzar consultes a les diferents taules. A més, es va prestar especial atenció en la inserció i eliminació d'incidències, ja que les dependències entre taules, obligaven a realitzar aquestes operacions de manera que es respectessin totes les relacions entre claus forànees. Per exemple, en el cas de les incidències de transport, hi ha 3 taules diferents que contenen informació sobre aquesta incidència. En una taula podem trobar atributs com latitud, longitud, causa, severitat, estat, TTL - Time To Live, data de l'última modificació i data d'enviament de l'incidència. En una altra taula, indexada per l'ID de la incidència, es guarda la descripció d'aquesta i l'identificador de l'usuari que ha reportat la incidència. Finalment, una tercera taula recull informació sobre el servei de transport públic al qual afecta la incidència amb atributs com servei, línia, estació i direcció.

A més, la informació sobre l'usuari es guarda en una altra taula, i la informació sobre l'estació es guarda en dues taules, mentre que en la taula `Queries` s'emmagatzema la informació relativa a les consultes realitzades. Així, per extreure tota la informació d'una incidència cal realitzar consultes sobre set taules diferents relacionades entre elles. De la mateixa manera, quan es vol afegir una incidència cal omplir sis taules diferents respectant les claus forànees.

La classe `GestorIncidències` s'encarrega de carregar un conjunt d'incidències segons els filtres seleccionats per l'usuari i de proporcionar informació genèrica sobre les incidències com el nombre de consultes a un determinat tipus d'incidència o servei de transport, mentre que la classe `GestorMapa` implementa un mètode que genera una seqüència XML amb la informació necessària de les incidències per localitzar-les en el mapa. Aquest XML es llegit i interpretat per un mètode implementat en JavaScript que, utilitzant funcions de l'API de Google Maps [8], permet crear el mapa i situar gràficament els markers. Aquest procés està esquematitzat en la figura 6, on es pot veure com servidor i client utilitzen el XML com a pont per comunicar-se la informació geo-posicionada, obtinguda pel servidor (utilitzant llenguatge PHP) connectant-se a la base de dades i llegida pel client (utilitzant llenguatge JavaScript) per mostrar a l'usuari el mapa amb les incidències geo-posicionades.

Cada cop que s'aplica un filtre, cal repetir aquest procés (esquematitzat en la Figura 6) de generació d'una seqüèn-

cia XML, que contindrà la informació només d'aquelles incidències que satisfacin els filtres seleccionats per l'usuari.

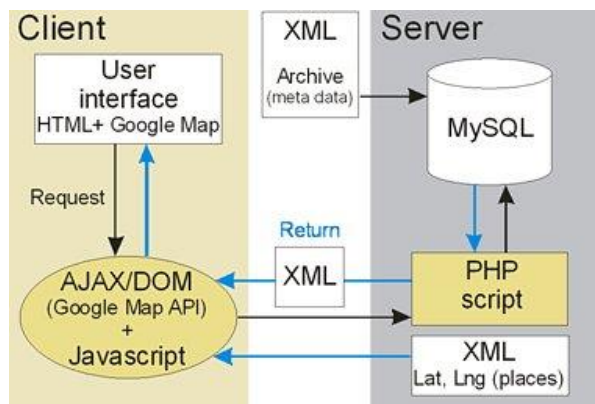


Figura 6: Esquema que mostra com generar i carregar marcadors en un mapa mitjançant l'API de Google Maps (Font: [www.isda.ncsa.uiuc.edu](http://www.isda.ncsa.uiuc.edu) )

La classe `ObjDB` afegeix un nivell d'abstracció a la base de dades, facilitant la connexió de les altres classes del model amb la base de dades. Aquesta classe implementa mètodes que utilitzen funcions de PHP per connectar-se o desconnectar-se amb la base de dades, realitzar consultes o extreure la informació de les consultes. D'aquesta manera, des de la resta de classes del model, utilitzant la classe `ObjDB`, es poden comunicar amb la base de dades sense necessitat de conèixer els mètodes específics de PHP. A més, en el cas de que en un futur, es volgués canviar el motor de base de dades, el llenguatge o els mètodes de PHP actuals quedessin desfasats, només caldria canviar la classe `ObjDB` i la resta de l'aplicació seguiria funcionant correctament.

Per al cas d'ús d'afegir una incidència de transport, a més d'omplir els camps corresponents (longitud, latitud, descripció, estat, severitat i causa) s'ha de triar l'estació i la direcció. Donat que les estacions canvien segons la línia, es va afegir la classe `Station` (que representa una instància de les taules relacionades amb les estacions de la base de dades `Incidències`), i la classe `GestorEstacions`, que conté el mètode `getStationsByLine`, amb el qual es recuperen totes les estacions d'una línia. Per tant, quan l'usuari canvia la línia, s'actualitzarà el llistat d'estacions a triar utilitzant una crida AJAX, a la vegada que es canviarà automàticament el servei (com ja es feia en el formulari per filtrar). La direcció és un camp numèric que pot tenir tres valors diferents: 0, 1 o 2. Si és 0 la direcció és desconeguda, si és 1 és la primera parada de la línia i si és 2 és l'última parada. Per tant, cal conèixer

En quant al mòdul de gestió de les T-11, s'ha dissenyat i implementat de manera molt similar al mòdul de gestió d'incidències, ja que els requisits eren els mateixos: veure un llistat de les T-11, veure un mapa i afegir, modificar i eliminar les T-11, amb la particularitat que com que no tenim diferents tipus de T-11, es simplifica molt més el disseny i la implementació.

Les classes `T11` i `GestorT11` es van crear per separat i sense cap tipus de relació amb les classes relacionades

amb les incidències, ja que en la base de dades d'Incidències hi ha moltes menys taules per enregistrar la informació de les T-11 i tenen molts atributs diferents. A més, així es permet tractar cada mòdul per separat, sense que un afecti l'altre. La classe T-11, doncs, permet carregar, afegir, modificar i eliminar una T-11, mentre que la classe GestorT11 recupera un conjunt de T-11 que satisfan unes condicions concretes.

Una altra funcionalitat de l'aplicació és la de generar informes de forma periòdica, els quals mostren informació estadística de l'ús de l'aplicació Incidències 2.0 i dels esdeveniments que han succeït durant un període determinat. Aquestes estadístiques recullen dades com el nombre d'usuaris actius, el nombre de consultes a l'aplicació, el nombre d'incidències reportades i consultades per cada tipus d'incidència.

Actualment ja es generen informes a partir d'un script en Python. El contingut dels informes que s'han de generar amb la nova aplicació és el mateix que el dels informes que es generen actualment a partir d'aquest script. Per tant, també s'han pogut aprofitar algunes consultes SQL que ja es realitzaven per generar els informes. Fins ara, aquest script generava els informes en LaTeX, mentre que ara l'aplicació els genera directament en PDF, ja que es un dels requisits del sistema. Les dades obtingudes es situen entre una data inicial fixada per l'usuari i la data actual.

La generació d'informes en format PDF s'ha dut a terme utilitzant la llibreria TCPDF [9] per a PHP. Aquesta llibreria porta incorporades diverses funcionalitats com ara generar el PDF, canviar el format del document, canviar l'encapçalament, el peu de pàgina o canviar el format del text. A més, permet escriure el PDF de dues formes diferents: mitjançant "cel·les" (mètode MultiCell), on tot el que s'escriuen són cel·les i taules que podem inserir movent-nos pel document segons dues coordenades X/Y, o escrivint directament codi HTML (mètode writeHTML), tot i que ha de ser HTML5 estricte i no estan disponibles tots els tags. També permet introduir certs estils CSS.

Donat que els documents per als informes han de tenir un format senzill, s'ha triat utilitzar la funció writeHTML per escriure el PDF, ja que requereix molt menys temps d'aprenentatge que els MultiCell i fa el codi molt més clar i entenedor de cara al programador.

Per construir l'informe es realitzen les consultes necessàries a la base de dades per separat. Després de cada consulta es construeix el codi HTML que s'ha d'escriure en el PDF amb el resultat obtingut de la consulta i el text a escriure, i s'envia a la classe TCPDF amb el mètode writeHTML. Un cop hem escrit tot el document, s'indica a la classe TCPDF que tanqui el fitxer, generant l'informe en PDF.

Finalment, es controla l'accés a l'aplicació utilitzant les sessions de PHP. Abans de poder accedir a l'aplicació, el sistema demanarà a l'usuari que s'identifiqui com a administrador.

L'usuari haurà d'introduir el nom d'usuari i el password corresponent al compte d'administrador de l'aplicació (prèviament introduït en la base de dades Incidències) per poder accedir-hi. L'aplicació valida aquestes

dades i, si corresponen al compte d'administrador, guarda en una variable de sessió un valor que es comprova sempre que es demana un recurs de l'aplicació.

En cas contrari, denega l'accés a l'aplicació, tornant a mostrar la pantalla d'identificació i informant a l'usuari que les dades que ha introduït no són correctes.

D'altra banda, totes les classes i mètodes han estat comentats seguint l'estàndar PHPDoc, una adaptació per a PHP de JavaDoc (una utilitat per la generació de documentació de APIs en format HTML a partir de comentaris en el codi font desenvolupat en llenguatge Java). Cal dir que avui en dia, PHPDoc encara és un estàndar informal, i que està en procés de formalització. El principal avantatge d'utilitzar aquesta eina és que permet generar documentació del codi directament a partir dels comentaris. A més, és una manera de comentar totes les classes i mètodes de la mateixa manera i seguint un mateix format. També s'han comentat aquelles línies de codi de mètodes i classes més complexes d'entendre, per tal de facilitar que, en un futur, altres desenvolupadors puguin treballar sobre el mateix codi més fàcilment.

## 4 EXPOSICIÓ DELS RESULTATS I PROVES

A continuació es presentaran els resultats obtinguts una vegada finalitzat el projecte i les proves que s'han realitzat per avaluar el compliment del requisits establerts i el correcte funcionament de l'aplicatiu.

### 4.1 Configuració de l'entorn

Tot i que existeix un entorn de producció d'Incidències 2.0 s'ha treballat sobre una còpia de la base de dades en un entorn de proves local per evitar possibles alteracions del sistema. Cal afegir que les dades amb les que s'han realitzat proves són dades reals copiades de la base de dades d'Incidències 2.0. D'aquesta manera ens assegurem que les dades i la informació que manipulem és exactament la mateixa que si féssim les proves en l'entorn real i així augmentem la confiança en els resultats de les proves.

Per tant, l'aplicació s'ha desenvolupat en un entorn LAMP (Linux + Apache + MySQL + PHP) [10]. Es va escollir aquest entorn per la seva condició de software lliure i la simplicitat que ofereix a l'hora de connectar els tres elements indispensables: codi, base de dades i servidor.

Es va instal·lar sobre una màquina amb sistema operatiu Ubuntu 13.10, un servidor web HTTP Apache, un sistema de gestió de bases de dades relacionals MySQL, un intèrpret de PHP i l'eina phpMyAdmin per facilitar l'administració de la base de dades.

El desenvolupament de codi s'ha fet sobre la plataforma Eclipse [11] amb el plugin PDT (PHP Development Tools), per facilitar la generació de codi i la gestió del sistema de fitxers de l'aplicació. Treballar sobre aquesta plataforma permet trobar errors de sintaxi que marca la pròpia plataforma, cercar text en el codi, cercar arxius, obrir la declaració d'un mètode o una variable des d'una part del codi on s'utilitza aquest mètode o aquesta varia-

ble i generar documentació amb l'utilitat PHPDoc (integrada en Eclipse).

## 4.2 Resultats

Finalment, el resultat final del projecte ha estat una aplicació web – la vista principal de la qual es pot veure a la Figura 7 – capaç de tractar amb informació geo-posicionada d'una base de dades, que utilitza diverses llibreries i tecnologies (Google Maps API, jQuery, Twitter Bootstrap, TCPDF i AJAX).

ID	Descripció	Estat	Data d'enviament	Línia	Servei
1	Per obres de millora de infraestructura la circulació dels trens de Rodalies es modifica de 23:20 d'avui fins les 7:20 de demà.	No vàlida		01	R
10	Circulació momentàniament aturada entre Ripoll i Ribes per obres a la via. S'està gestionant un servei alternatiu per aquest tram.	No vàlida		04	R
100	Tren Manresa 16:30 a Vilafraça 18:38 circula amb 20 minuts de retard per incidència a la via. S'està gestionant un servei alternatiu per aquest tram.	No vàlida		04	R
101	Tren Hospitalet 19:13 a Vic 20:46 circula amb 20 minuts de demora per congestió a l'entorn de Vilanova del Valles.	No vàlida		04	R
102	Per atropellament, tren La Tor Querol (11:15) a l'Hospitalet (14:14) aturat a l'entrada de Montcada Ripollit.	No vàlida		04	R
103	Per atropellament entre Montcada Ripollit i Sant Perpetuà, circulació interrompuda en aquest tram.	No vàlida		04	R
103822114		No vàlida	2012-02-13 22:37:51	04	R
104	Normalitzada la circulació dels trens entre Montcada Ripollit i Santa Perpètua. Els trens aniran recuperant el seu horari normal de pas.	No vàlida		04	R
105	Tren de la Tour de Carol (17:00) a l'Hospitalet (20:20) circula amb 20 minuts de demora.	No vàlida		04	R
106	Tren de Vic (20:10) a l'Hospitalet (21:21) detingut a l'estació de Montcada Ripollit a l'espera d'assistència sanitària.	No vàlida		04	R

Figura 7: Pantalla principal de l'aplicació web, que mostra el llistat d'incidències

L'aplicació extreu informació d'una base de dades i la tracta per mostrar-la a l'usuari de formes gràfiques diferents (llistat, mapa i PDF), i permet a l'usuari cercar informació mitjançant l'aplicació de filtres i realitzar operacions sobre aquesta base de dades (els resultats visuals de l'aplicatiu es poden veure en l'Annex A3).

La durada total del projecte ha estat de 300 hores de treball, tenint en compte totes les tasques de planificació, anàlisi, reunions, documentació, implementació i proves.

Per assolir els objectius, es van identificar les tasques i activitats a realitzar i es va estimar la durada aproximada de cada una. Segons les tasques identificades i el calendari del projecte, es va fer una planificació en el temps del projecte, representada per un diagrama de Gantt i es van marcar cinc fites a assolir.

Després del mòdul de gestió d'incidències es va haver de fer un reajustament de la planificació, donat que es va trigar una setmana més del compte a completar aquest mòdul. Aquest retràs es va produir degut a que, en la planificació inicial, es va estimar que la duració del temps per la formació amb les noves tecnologies i eines (especialment l'API de Google Maps i Twitter Bootstrap), per provar l'entorn de proves i per la implementació del mòdul era menor de la que va resultar realment.

Aquest retràs, però, es va veure compensat pel reaprofitament de codi en el mòdul de gestió de les T-11, i perquè es van utilitzar moltes classes i mètodes implementats durant el desenvolupament del mòdul de gestió d'incidències (com la classe que comunica amb la base de dades).

## 4.3 Proves realitzades

Les classes del model s'han provat a partir de la definició de casos de test, fent crides als mètodes de cada classe amb diferents inputs i comprovant que els outputs eren els esperats. Per tant, cada classe ha estat testejada per un joc de proves abans de ser utilitzada en els controladors.

Un cop implementat cada submòdul, s'ha provat per assegurar que es compleixen els objectius establerts i que fa la funcionalitat demanada a partir de tests de funcionalitat. A més, s'han fet proves d'integració per comprovar que els diferents submòduls segueixen funcionant després d'incorporar noves funcions a l'aplicació.

Per als tests de funcionalitat i d'integració s'ha utilitzat la tècnica Exploratory Testing [12], que consisteix a provar totes les funcionalitats del software, considerant tants escenaris com sigui possible, tant aquells on l'aplicació hauria de respondre fàcilment com els que no és tant evident que respongui de forma correcta, a més de provar diferents combinacions d'escenaris, per trobar comportaments ocults i incorrectes de l'aplicació.

Aquesta tècnica, a més, es basa en la idea de que explorant el software, a més de realitzar proves, aprenem a utilitzar-lo, investiguem per trobar la manera d'aconseguir el comportament de l'aplicació que desitgem, estudiem les diferents seqüències de passos que hem de dur a terme per a que l'aplicació realitzi certa acció, descobrim (molts cops per casualitat) noves accions o comportaments no esperats i, en definitiva, guanyem coneixement sobre l'aplicació, la qual tractem com una caixa negra amb la que investiguem i explorem.

Com es mostra en la Figura 8, el conjunt de passos de l'Exploratory Testing forma un circuit tancat, en que sempre es poden aprendre nous coneixements sobre el comportament del software després de l'anàlisi dels resultats del test.



Figura 8: Seqüència de passos de l'Exploratory Testing (Font: [www.staqs.com](http://www.staqs.com) – Software Testing and Quality Services)

En el cas del mòdul de gestió d'incidències, l'Exploratory Testing ha estat especialment efectiu a l'hora de provar el sistema de filtratge, ja que molts filtres anaven per separat, però deixaven de funcionar correctament combinant-los entre ells o aplicant certs filtres abans que altres. De la mateixa manera ha servit per assegurar que el mapa es carregava sempre correctament, ja



que l'API de Google Maps es especialment delicada en entorns dinàmics i, en ocasions, després d'accedir a una altra pantalla, el mapa es desquadra, desapareix o no carrega bé els marcadors. Per això ha calgut dedicar molt temps a provar aquesta funcionalitat de l'aplicació i corregir tots aquests errors.

Un altre aspecte delicat de l'aplicació són les DataTables. Les DataTables són elements incorporats en el framework jQuery que permeten construir taules de dades amb diverses funcionalitats incorporades, com per exemple paginació segons el nombre d'elements per pàgina que selecciona l'usuari o ordenar els elements de la taula segons una columna. Aquestes taules són especialment sensibles quan es manipula informació de la base de dades. Com que en la nostra aplicació totes les operacions que realitzem repercuteixen directament en les DataTables (aplicació de filtres, afegir incidències, modificar incidències i eliminar incidències), cal recarregar-les cada cop que l'usuari realitza una acció. Durant les proves s'ha de verificar que les DataTables reflecteixen els canvis produïts per les accions de l'usuari correctament i que no es desconfiguren.

Un cop provats tots els mòduls per separat, s'han realitzat proves d'integració, comprovant que no s'havia modificat el comportament del mòdul de gestió d'incidències. Un dels bugs trobats va ser que, al integrar dues DataTables (necessàries tant per al llistat d'incidències com per al llistat de les T-11) i dos mapes de Google Maps es produïen conflictes entre mòduls, i apareixien errors de JavaScript, que fan que les taules es desconfiguren o deixin d'utilitzar la paginació. Aquests bugs trobats gràcies als testos d'integració van ser solucionats en la fase final del projecte.

També s'ha utilitzat l'eina Firebug, incorporada en el navegador Firefox, per assegurar que no apareixen warnings de JavaScript i que totes les peticions HTTP es realitzen correctament, ja que podrien aparèixer errors interns que no veiem a simple vista a mesura que l'aplicació creix.

Amb aquesta eina s'han pogut trobar i s'han corregit errors com per exemple ús de variables no declarades de JavaScript o errors produïts per recursos no trobats, que a simple vista no influeixen en el correcte funcionament del software, però que podrien ocasionar errors si no es tracten. De la mateixa manera, també s'han suprimit tots els warnings que originava el codi PHP, per evitar comportaments no desitjats del model i els controladors.

## 5 CONCLUSIONS I LÍNIES FUTURES

A continuació s'exposen les conclusions del projecte i les possibles línies de millora.

### 5.1 Conclusions

Durant aquest projecte s'ha desenvolupat un aplicatiu web que permet explotar de forma eficient la informació geo-posicionada generada a partir de l'aplicació Incidències 2.0, permetent als seus administradors consultar informació, modificar i eliminar registres de la base de da-

des i generar estadístiques per un període de temps concret.

S'han complert, doncs, tots els objectius establerts en la planificació inicial en el temps previst i amb els recursos disponibles identificats en la planificació inicial.

A l'hora de començar un nou projecte, es plantegen problemes que cal resoldre utilitzant tecnologies o eines no conegudes i que requereixen un temps de formació, que cal tenir en compte en la planificació per tal d'evitar retrassos que forcen canvis en la planificació, com ha passat en aquest projecte.

En les fases inicials de la implementació el temps pot ser molt elevat, però si el disseny es bo i es genera codi de qualitat (reutilitzable, comentat, seguir els patrons de disseny, etc.), el temps d'implementació és molt baix per a fases posteriors, ja que és molt fàcil aprofitar codi i funcions ja implementades.

Es possible desenvolupar software de qualitat en poc temps si es segueixen conceptes de metodologies àgils de desenvolupament i es posa l'èmfasi adequat en cada etapa del projecte, sense obviar cap fase de l'enginyeria del software i seguint una metodologia adient.

A més a més, ha estat un punt clau del projecte dedicar un temps a planificar, pensar, dissenyar i realitzar les proves, ja que en aplicacions on hi ha molts elements que depenen els uns dels altres, se'ns poden passar per alt comportaments no desitjats que apareixen fent una seqüència d'accions concreta.

Finalment, es pot concloure que amb 300 hores es pot desenvolupar un projecte de desenvolupament de software de forma satisfactòria, aplicant la metodologia adient i les eines necessàries. No obstant, han quedat molts punts pendents per a possibles línies futures. Es una realitat, doncs, que el desenvolupament d'un projecte de qualitat en l'àmbit de l'enginyeria informàtica requereix moltes hores d'estudi, formació, planificació, anàlisi, disseny, implementació i proves, com passaria en qualsevol altre projecte d'enginyeria.

### 5.2 Línies futures

Tot i que s'ha desenvolupat la funcionalitat bàsica de l'aplicació, encara hi ha una sèrie de futures millores que es podrien aplicar un cop tancat el projecte.

Una d'elles seria millorar el sistema d'autenticació i accés a l'aplicació, utilitzant tecnologies més segures, com ara els tokens. Caldria, doncs, estudiar i implementar alternatives que s'utilitzen en l'actualitat com els JWT [12] que permeten implementar un sistema d'autenticació amb tokens [13] i seguretat de clau pública [14].

Una altra possible millora es la inserció de gràfics i taules en els informes generats per l'aplicació. Caldria estudiar si es possible incloure gràfics en els PDFs amb la llibreria TCPDF o si s'hauria de canviar de llibreria.

També es podria revisar si l'aplicació compleix l'arquitectura REST [15] i redissenyar aquells components de l'aplicació que no segueixen aquesta arquitectura, per mirar de fer que l'aplicació sigui RESTfull, es a dir,

que compleixi totalment les bases de REST.

A més a més, l'aplicació ha estat preparada per a poder ser internacionalitzada fàcilment, afegint tants idiomes com es desitgi. Una bona millora seria traduir els textos a altres idiomes.

Un cop s'hagin introduït aquestes millores, es podria llençar una versió de prova que sigui provada per usuaris reals durant un temps. Després de categoritzar i prioritzar els possibles bugs trobats per l'usuari i els canvis demanats, s'hauria de valorar quins canvis introduir i com tractar els bugs trobats.

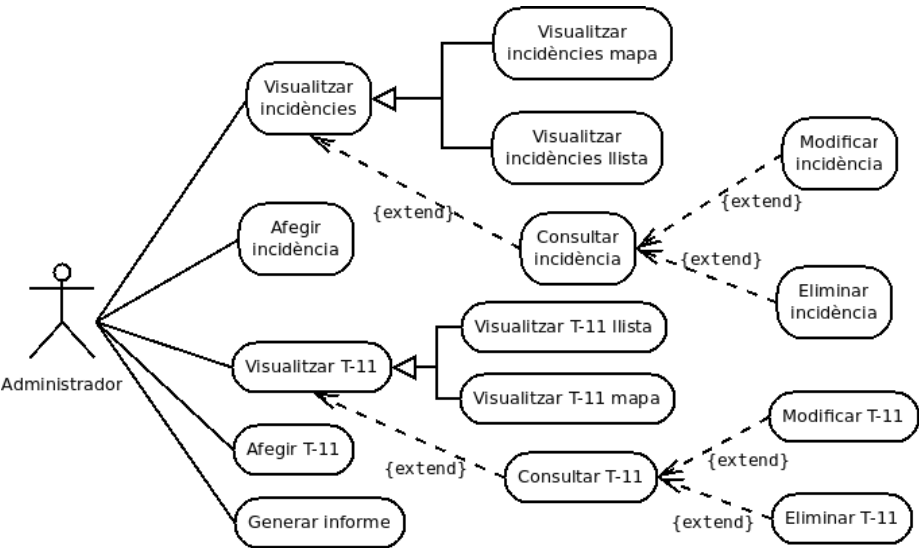
Finalment, caldria provar l'aplicació en l'entorn de producció real i integrar-la dins del projecte Incidències 2.0.

## BIBLIOGRAFIA

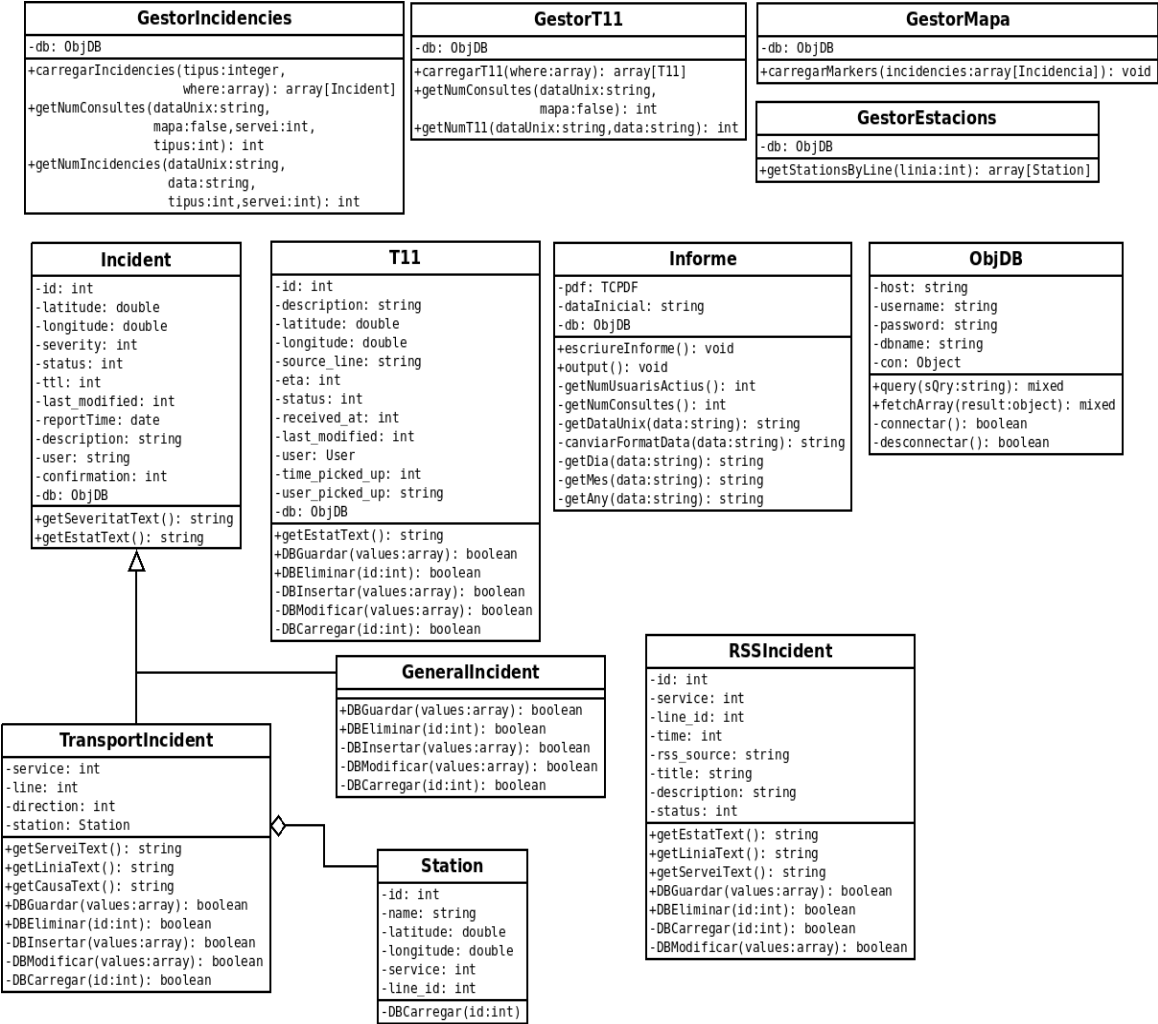
- [1] *Web Incidències 2.0*. <http://www.incidencies.org> (consultat el 16 de març de 2014).
- [2] Beaudouin-Lafon, Michel; Mackay, Wendy E. *Prototyping tools and techniques*. Université Paris-Sud. <https://www.lri.fr/~mackay/pdffiles/Prototype.chapter.pdf>
- [3] Gordon, V. Scott; Bieman, James M. *Rapid Prototyping: Lessons Learned*. Department of Computer Science, Colorado State University, 1997.
- [4] Parr, Terence. *Enforcing Strict ModelView Separation in Template Engines*. University of San Francisco, 2004.
- [5] *Documentació PHP5*. <http://www.php.net/manual/es/> (consultat el 16 de març de 2014).
- [6] Skinner, Jesse. *Unobstrusive Ajax*. O'Reilly Shortcuts, 2007. ISBN: 978-0-596-51024-4.
- [7] Powell, Thomas. "The XMLHttpRequest Object". *AJAX: The Complete Reference*. Mc Graw Hill, 2008. ISBN: 978-0-07-149216-4. 99-152.
- [8] *Documentació Google Maps API*. <https://developers.google.com/maps/?hl=es> (consultat el 16 de març de 2014).
- [9] *Documentació llibreria TCPDF per a PHP*. <http://www.tcpdf.org/docs.php> (consultat el 27 de maig de 2014).
- [10] Boronczyk, Tim. *Beginning PHP6, Apache, MySQL web development*. Indianapolis, IN: Wiley Pub, 2009.
- [11] *Base de dades de bugs d'Eclipse*. <https://bugs.eclipse.org/bugs/> (consultat el 16 de març de 2014).
- [12] Bach, James. *Exploratory Testing Explained*, 2003.
- [13] Jones, Michael, Bradley, John, Sakimura, Nat. *JSON Web Token (JWT)*. Internet Engineering Task Force (IETF), 2014.
- [14] EdgeCast Networks Inc. *Token Based Authentication Administration Guide*, 2010.
- [15] Menezes, A., Van Oorschot, P., Vanstone, S. "Public-Key Encryption." *Handbook of Applied Cryptography*. CRC Press, 1996. 283-312.
- [16] Enterprise Applications Division of the Systems and Network Analysis Center (SNAC). *Guidelines for Implementation of REST*. National Security Agency, 2011.

ANNEX

A1. Diagrama de casos d'ús




A2. DIAGRAMA DE CLASSES



A3. RESULTATS VISUALS DE L'APLICATIU

Pantalla principal



Incidents 2.0

admin

Idioma

Incidents

T-11

Informes

Filtres

Tipus d'incidència

Estat

Severitat

Servei

Linia

Causa

Incidències

Llistat

Mapa

Afegir incidència

10

entrades per pàgina

ID	Descripció	Estat	Data d'enviament	Linia	Servei
1	Per obres de millora de infraestructura la circulaci?? dels trens de Rodalies es modifica de 23:20 d'avui fins les 7:20 de diumenge.	Caducada		R1	R
10	Circulaci?? moment??niament aturada entre Ripoll i Ribes per obst??cies a la via. S'est?? gestionant un servei alternatiu per aquest tram	Caducada		R3	R
100	Tren Manresa 16:30 a Vilafranca 18:38 circula amb 20 minuts per incid??ncia t??cnica del tren ja resolta	Confirmada		R4	R
101	Canviem!!	Sense confirmar		R3	R
103	Per atropellament entre Montcada Ripollet i St Perpetua, circulaci?? interrompuda en aquest tram.	Caducada		R3	R
1036021114	eeeeep!!!	Confirmada	2012-02-13 22:37:51	R4	R
104	Normalitzada la circulaci?? de trens entre Montcada Ripollet i Santa Per??tua. Els trens	Caducada		R3	R

Llistat d'incidències

Incidències

Llistat

Mapa

Afegir incidència

10

entrades per pàgina

ID	Descripció	Estat	Data d'enviament	Linia	Servei
1036021114	eeeeep!!!	Confirmada	2012-02-13 22:37:51	R4	R
1060311331	provant provant	Sense confirmar	2011-06-27 15:30:34	R7	R
1074150553	🚗/j 🚗🚗	Caducada	2012-02-16 17:58:49	R1	R
112626334	tren descarrilat	Confirmada	2012-02-09 11:13:06	R1	R
1126766833	🚗🚗e	Caducada	2012-02-17 11:18:02	R3	R
1181550447	Prova	Caducada	2012-02-16 11:39:25	R3	R
122414478	millor descripció	Caducada	2012-02-08 20:06:43	R1	R
1282073806	🚗🚗e	Caducada	2012-02-13 23:14:30	R1	R
1288271586	servei aturat per una incidencia	Caducada	2012-02-07 19:55:28	L4	🚗
1316347572	Prova 1 (Cristian)	Caducada	2014-01-10 12:00:16	R1	R

Mostrant 1 a 10 de 85 entrades

← Anterior

1

2

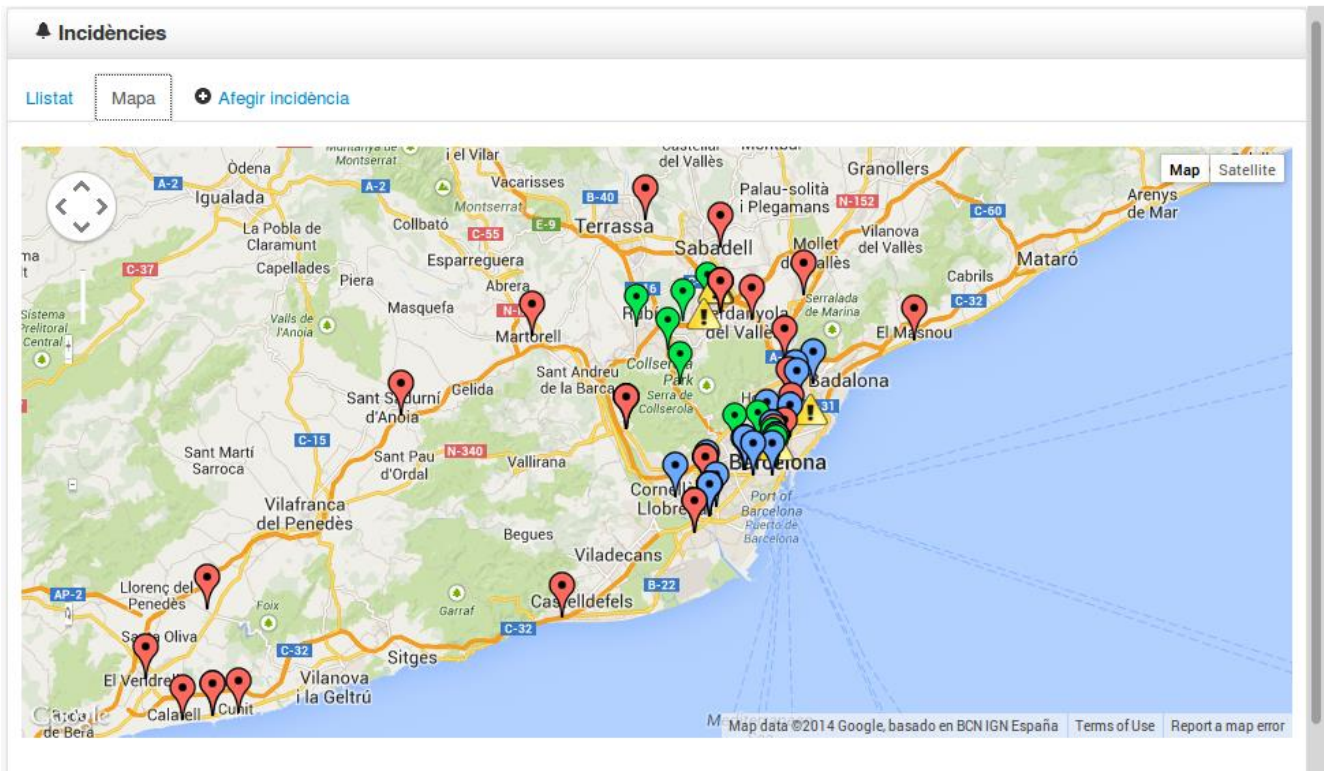
3

4

5

Següent →

Mapa d'incidències



</



Afegir incidència

Incidències

Llistat

Mapa

Afegir incidència

Nova incidència de transport

Latitud

0

Longitud

0

Causa

Retard

Severitat

Alta

Estat

Sense confirmar

Servei

Rodalies

Linia

R1

Estació

Molins de Rei

Direcció

Desconeguda

Confirmació

Sense confirmar

TTL

50

Descripció

Escriu aquí la descripció

Cancelar

Guardar canvis

Llistat T-11

Incidències

T-11

Informes

Filtres

Estat

T-11

Llistat

Mapa

Afegir T-11

10

entrades per pàgina

ID	Descripció	Estat	Linia	Data d'enviament	Data de recollida
1043234618	paperera5	Confirmada	metro L3	09-02-2012 22:57:48	09-02-2012 22:58:36
1175760851	☞☞e	Caducada	☞☞e	17-02-2012 13:03:06	01-01-1970 01:00:00
1308844391	dlddfd	Sense confirmar	R2 Nord	08-02-2012 22:49:52	08-02-2012 22:50:14
1348939880	☞☞e	Caducada	☞☞e	13-02-2012 22:32:31	13-02-2012 22:35:50
136406114	gelador	Caducada	jutjant	08-02-2012 22:52:09	01-01-1970 01:00:00
1426928830	☞☞e	Caducada	☞☞e	17-02-2012 13:00:05	17-02-2012 13:00:52
1441305288	☞☞e	Caducada	☞☞e	17-02-2012 13:05:57	01-01-1970 01:00:00
1444728255	a la paperera	Caducada	L5	19-02-2012 00:36:03	19-02-2012 00:36:37
1456821665		Caducada		08-02-2012 10:14:53	08-02-2012 10:15:13
150916066	al banc	Caducada	r4	09-02-2012 10:35:07	01-01-1970 01:00:00

Mostrant 1 a 10 de 71 entrades

← Anterior

1

2

3

4

5

Següent →

## Generar informes

[Incidències](#) [T-11](#) [Informes](#)

Data inicial de l'informe

[Generar informe](#)

June 2014

Su	Mo	Tu	We	Th	Fr	Sa
		3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

## Informe en PDF

**Informació setmanal de la BBDD d'Incidències 2.0****Jun 29, 2014**

En aquest document es presenten les dades obtingudes entre el dia 04-01-2011 i el dia 29-06-2014 (ambdós inclosos).

**1. Usuaris actius**

Considerarem els usuaris actius aquells que han realitzat almenys una consulta en el període analitzat.

- Nombre usuaris actius: 741
- Número de consultes a l'aplicació: 7125

**2. Incidències transport públic****2.1 Dades agregades**

Dades conjuntes de totes les línies de transport (RENFE, FGC, Metro).

- Nombre d'avísos RSS publicats: 198
- Nombre d'incidències transports públics: 85
- Nombre de consultes d'incidències transports públics: 2039
- Nombre de consultes d'incidències transports públics amb el mapa: 6172

**2.2 RENFE**

- Nombre d'incidències de RENFE: 49
- Nombre de consultes d'incidències de RENFE: 1899

**2.3 FGC**

- Nombre d'incidències de FGC: 12
- Nombre de consultes d'incidències de FGC: 748